UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) BRL-tr-2908	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION US Army Ballistic Research Laboratory — 6b. OFFICE SYMBOL (If applicable) SLCBR-IB	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code) Aberdeen Proving Ground, MD 21005-5066	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION — 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. 61102A; PROJECT NO. AH43; TASK NO.; WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

AN EVALUATION OF FIVE PC-BASED EXPERT SYSTEM SHELLS

12. PERSONAL AUTHOR(S) Frederick J. Shaw and Robert A. Fifer

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM Sep 86 TO Mar 87		

16. SUPPLEMENTARY NOTATION

17. COSATI CODES FIELD	GROUP	SUB-GROUP
09	02	
21	02	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Expert Systems, Expert System Shells, Artificial Intelligence, Propellant Formulations, Personal Computers, INSIGHT 2+, KDS, TIMM, 1st-Class, EXSYS

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Five expert system shells (three example-based and two rule-based) were evaluated for this project. These shells included KDS (KDS Corporation), 1st-CLASS (Programs In Motion Inc.), TIMM (General Research Corp.), EXSYS (EXSYS Inc.), INSIGHT 2+ (Level Five Research Inc.). Each of these expert system shells offered advantages and disadvantages. The example-based systems offer the advantage of being easier to learn to use; all that is needed is to enter examples and have the shell develop the rules from the data. These shells ~~also~~ offer an~~other~~ advantage in that they can help reveal interrelationships that are not readily apparent, since they draw their own conclusions from the data. However example-based systems are more rigid and inflexible which makes them more difficult to work with. Rule-based systems are more difficult to learn to use, since they involve entering if-then-else rules. However (especially in the case of INSIGHT 2+) they offer more flexibility and freedom, since the knowledge engineer or programmer has control over the development and use of the rules.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL MR. FREDERICK J. SHAW	22b. TELEPHONE (Include Area Code) 301-278-6167 — 22c. OFFICE SYMBOL SLCBR-IB-I

DD Form 1473, JUN 86 — Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

19. Abstract (Cont'd):

Based on the results of this evaluation, the propellant formulation expert system will be developed using the expert system shell, INSIGHT 2+. A few of the more important features of INSIGHT 2+ are that it uses a programming language, the ease with which TURBO PASCAL programs and dBASEIII databases can be accessed. INSIGHT 2+, for this task, proved most advantageous. However, for other applications, which shell to use is highly dependent on the type of expert system being developed and the knowledge engineer's personal preferences.

TABLE OF CONTENTS

LIST OF TABLES

I. INTRODUCTION

Artificial Intelligence (AI) encompasses many different aspects of computer technology. A few of the major sub-disciplines are expert systems, robotics, and natural language processing. In recent years, expert systems have been applied to an increasing number of commercial and government laboratory applications. An expert system is computer software which simulates the knowledge and reasoning power of a human expert. The computer program queries the user for the necessary data and then based on the information received, the program gives a suggestion/solution to the problem. Some more famous examples of expert systems are:

1. R1 - An expert system developed at DEC for use in configuring computer orders.[1]

2. DENDRAL - An expert system which interprets mass spectra to determine molecular structures.[2]

3. MYCIN - An expert system which diagnoses and then suggests the treatment for specific blood diseases.[3]

Expert systems consist of two parts, the "knowledge-base" and the "inference engine". MYCIN was one of the first expert systems in which the inference engine was extracted or separated from the knowledge-base. The inference engine, EMYCIN (extracted MYCIN), could then be used with other knowledge-bases. This led to expert system tools which help in the development of expert systems. These tools offer built-in capabilities like debugging aids, input/output facilities, explanation facilities, and knowledge-base editors. These facilities help enhance the expert system, in addition to reducing the development time of the expert system.[4,5]

One type of expert system tool which has recently become commercially available are expert system "shells". These shells can be categorized as either example-based or rule-based. For example-based shells, the shell develops the expert system based on examples entered by the knowledge engineer or programmer. The shell itself develops the rules to be used in the expert system. On the other hand, rule-based systems require the knowledge engineer to enter the rules that are to be used by the expert system. Rule-based shells offer more flexibility and freedom, but these systems are more difficult to learn to use, since the knowledge engineer must first develop the rules.[6]

Since personal computers (pc's) have become increasingly powerful, many applications that before could only be run on mini/mainframe computers are now being developed on personal computers. One important area that has moved to the personal computer domain is expert systems and expert system shells. There are a considerable number of expert system shells available. Most of these shells are designed for use on personal computers; however, a few of these shells have versions which run on both mini/mainframe computers and personal computers. Also available are shells which are based on AI-computers, i.e., computers (pc's and mini's) which are designed specifically for using LISP.[7] Appendix A shows a current listing of pc-based shells.

The Ignition and Combustion Branch has been tasked to develop a propellant formulation expert system. The goal of the project is to develop an expert system for propellant formulation design. The expert system will aid the user in designing a propellant for an intended application, given certain user-specified constraints. The results of exercising the expert system will be a list of possible formulations (combinations of oxidizer, polymer, plasticizer, additives, etc.), as well as estimated propellant properties (energy, burning rate, sensitivity, etc.). Examples of user-specified constraints would be cost, and any ingredient preferences. Rules will be used both to select appropriate combinations of ingredients, as well as to estimate the properties of the resulting propellant formulations. For this project five expert system shells (three example-based and two rule-based) were evaluated; these included KDS (KDS Corporation),[8] 1st-CLASS (Programs In Motion Inc.),[9] TIMM (General Research Corp.),[10] EXSYS (EXSYS Inc.),[11] and INSIGHT 2+ (Level Five Research Inc.).[12]

II. EVALUATION

An expert system, which was analogous to the propellant formulation system to be developed, was used to evaluate the five expert systems shells to determine which would be the most suitable. The expert system shells were evaluated on: (1) ease of use in developing the expert system; (2) ability to pass data between an external program and the expert system; and (3) the ability to search databases. The shells purchased varied considerably in these three categories. The evaluations were based on procedures which were considered to be applicable to the development of the planned propellant formulation expert system. The version of TIMM evaluated was based on a VAX/VMS minicomputer. The personal computer version of TIMM is identical to the VAX version except for the memory limitations. The other four expert system shells were run on a WYSEpc XT-compatible PC with 640k memory, and on a Zenith ZWX-248 AT-compatible PC with 640k memory + 2560k extended memory and 80287 math coprocessor.

A. KDS

Knowledge Delivery System (KDS) is an example-based shell. Version 2.0 which was purchased for this evaluation is not capable of performing mathematical calculations within the shell program. (NOTE: The updated version 3.0 is capable of performing mathematical calculations within the shell program.) The KDS manual is very vague about the type of algorithm used in the reasoning process. KDS does offer the user the option of selecting either forward chaining or backward chaining.

In forward chaining, the first rule in the knowledge-base is executed. From this rule's premise the next rule containing this premise or a fact supporting this premise is executed. This cycle is continued until all the goals are proved or disproved. For backward chaining, the shell starts with a goal and then searches backwards for a rule containing a premise supporting this goal. The rules are then searched to find a fact which verifies this premise. If a fact is not found, a search is made for a rule that can be used to infer the fact or else the shell queries the user. Just as in forward chaining, this cycle is continued till the goal is proved, disproved, or all rules have been evaluated. If backward chaining is desired, the user selects

a goal to evaluate. If the goal selected is not proved or disproved then the shell reverts to forward chaining to reach a conclusion.

KDS's user interface is called an "intelligent development environment". The user enters representative examples, which KDS then uses to develop its own if-then-else rules. KDS interrogates the user just like the expert system would. What follows is a typical session showing how an expert system is developed:

```
previously entered:  condition 0
     "application is rocket" condition 0
     if true "metal AP" case 0
     if false "cat db" case 1

new case: ammonium perchlorate
   "KDS": application is rocket condition 0
   Response:  [y]
   "KDS": answer: metal AP case 0
   Response:  [n]
   "KDS":  what is the correct answer?
   Response:  ammonium perchlorate case 2
   "KDS":  What condition if true would distinguish case 0 from case 2 ?
   Response:  cost is moderate condition 1
```

This dialogue would continue until all the cases had been entered. One major problem with this is that only one condition can be entered per case, so that one has to go back and use the utility "fill in all conditions" for the remaining cases. Also each condition is considered unique so we are not able to have the shell exclude certain conditions based on the answer received. For example if "the development time is long" was set to true then "the development time is short" should therefore be set to false, yet we were not able to program the shell to do this. KDS's technique of having the user enter cases, and conditions which back up these cases is interesting, and as far as we know unique to expert system shells in general. Once all the cases are entered, the expert system is compiled. During the compilation process, if-then-else rules are generated and can be printed if desired. A nice feature that KDS offers is that text can be associated with specific conditions, therefore a more detailed explanation can be displayed, if needed. This text can be displayed automatically or only when requested by the user. Special cases, "knockout cases", can be entered that help exclude impossible situations or narrow down the searching process. These are similar to TIMM's exclusionary rules which are described below.

Another feature of KDS is that modules can be chained together. This is especially useful when developing a large expert system since the expert system can be written in smaller modules which are then chained together. Modules can be chained to conclusions, so that if a specific conclusion is reached an associated module will be called. In addition to conclusions, modules can also be chained to conditions. These features enable the knowledge engineer to use both forward chaining and backward chaining. For example to use the forward chaining mode, a module would be chained on a specific conclusion. Therefore depending on the conclusion reached an associated module would be invoked. For backward chaining a module would be

chained on a condition. Therefore if the value of a specific factor was needed then the associated module would be invoked.

KDS does not support the use of confidence factors; instead "a proprietary algorithm for resolving uncertainty using 'don't care' is used instead".[8] External programs can be called from within the KDS shell. The data is passed to the program as command-line arguments. Data is passed back to the shell using an interrupt vector and a driver program, which interfaces the external program to the shell. There are no special provisions for searching databases. The knowledge engineer has two choices: use an external program to access the database and then pass the data back to the shell, or input the data into the shell as case examples. This shell was written in assembly language, and therefore is fast and efficient.

B. 1st-CLASS

1st-CLASS (version 2.04) is another example-based expert system shell. It uses a spreadsheet-type format for the user interface. The factors are entered with their allowed values. An example of this is shown in Table 1. Once all examples are entered, the system generates a single rule in the form of a challenge tree as can be seen in Table 2. This rule is then used for querying the user and the tree is traversed based on the answers received. Two alternatives to using the rule generated by the shell is to either develop a unique rule by hand or to use the matching facility which searches all examples for a match.

Table 1. 1st-CLASS Examples

	Nitramine	Binder	Plasticizer	Result	Weights
1:	RDX	r45m/idpi	none	short(<10)	[1.00]
2:	HMX	r45m/idpi	none	med(10-20)	[1.00]
3:	RDX	gap	bttn	short(<10)	[1.00]
4:	HMX	gap	bttn	med(10-20)	[1.00]
5:	HMX	bammo/ammo	bttn	short(<10)	[1.00]
6:	HMX	bammo/ammo	tmetn	long(21-35)	[1.00]
7:	HMX	gap	tmetn	med(10-20)	[1.00]
8:	RDX	gap	tmetn	long(21-35)	[1.00]
9:	HMX	bammo/ammo	none	vlong(>35)	[1.00]
10:	RDX	r45m/idpi	none	short(<10)	[1.00]
11:	RDX	gap	none	vlong(>35)	[1.00]
12:	none	gap	bttn	vlong(>35)	[1.00]
13:	none	gap	tmetn	vlong(>35)	[1.00]

1st-CLASS's rule building process is based on Quinlan's ID3 algorithm.[13] 1st-CLASS classifies the data into classes or categories. Rules are developed which consist of decision trees developed from the data classes and from the conclusions. This technique is called inductive classification, since the rules are induced from the classification of the data.

The 1st-CLASS version evaluated was not capable of performing simple mathematical calculations within the shell. However an updated version does have this provision. The shell does recognize greater than, less than, and number ranges. These provisions are critical when searching databases for

numeric values, since it is usually not necessary to have an exact match, but only sufficient to know that a value is within certain limits. Some sample programs showing how to pass data to and from external programs (written in TURBO PASCAL) were included as examples. The shell passes data on the command-line to the external program. To pass data back to the shell, the external program is given an address in memory where the data is to be put. The external program has a choice of how to find this address. It can be passed as a command-line argument, or it can be retrieved from a fixed address in memory.

Table 2. 1st-CLASS Challenge Tree Rule

```
---- start of rule ----
nitramine??
/rdx:plasticizer??
|    |none:binder??
|    |     |r45m/idpi:-----------------------------------------short (<10)
|    |     |gap:-----------------------------------------------vlong (>35)
|    |     |bammo/ammo:----------------------------------------no data
|    |     |bammo/thf:-----------------------------------------no data
|    |tmetn:---------------------------------------------------long (21-35)
|    |bttn:----------------------------------------------------short (<10)
| HMX:binder??
|    |r45m/idpi:-----------------------------------------------med (10-20)
|    |gap:-----------------------------------------------------med (10-20)
|    |bammo/ammo:plasticizer??
|    |           |none:----------------------------------------vlong (>35)
|    |           |tmetn:---------------------------------------long (21-35)
|    |           |bttn:----------------------------------------short (<10)
|    |bammo/thf:-----------------------------------------------no data
|none:---------------------------------------------------------vlong (>35)
---- end of rule ----
Active examples: 13; Answer's examples: 2; Examples: 1, 10
```

As with the other shells 1st-CLASS supports chaining of modules together. An additional feature of 1st-CLASS is that separate modules can be chained to factors or results. These features enable the knowledge engineer to use both forward chaining and backward chaining. For example to use 1st-CLASS in the forward chaining mode, a module would be chained to a specific conclusion. Therefore depending on what conclusion was reached, a specific module would be called. For backward chaining a module would be chained to a condition or factor. Therefore if the value for a specific factor was needed then the associated module would be called. Special capabilities for I/O file operations (e.g., being able to import and export text files for use as data examples) are included. This is convenient because data from, for example, a database could be written to a text file and then evaluated by 1st-CLASS.[9]

C. TIMM

The Intelligent Machine Model (TIMM) is the third example-based expert system shell evaluated. The expert system is started by first entering the factors, their associated values, and the possible conclusions that the expert system can reach. This is a characteristic of TIMM which is similar to 1st-

CLASS since both require the user to enter the factors and their values first. TIMM develops it's conclusion based on "the least rectilinear distance" to the nearest neighbor. The reliability of the conclusion is calculated from the distance to the nearest neighbor. This reasoning process is called one nearest neighbor (1NN).

The factor values can be unordered (i.e., random), linearly ordered (i.e., 1,2,3,4), circularly ordered (i.e., the values cycle: Saturday, Sunday, Monday). There is an option for associating text to a factor which then can be requested by the user to get a more detailed explanation. An example of TIMM's "decision structure" for an expert system is shown in Table 3. The expert system is developed by giving solutions to the rules. In Table 4 one can see that the "if" part of the rule is factors and their corresponding values, the "then" part is the solution to the rule. There is also available a secondary knowledge-base which contains exclusionary rules, i.e., rules for impossible situations. These help in narrowing down the possible combinations of factor values, and therefore reduce the number of rules needed. The use of "greater than", and "equal to" in conjunction with ordered factors, also greatly reduces the number of possible combinations. In Table 4, rule 1S is an example of an exclusionary rule, and rule 38 is an example of a rule using "greater than or equal to" for an ordered factor. A knowledge engineer has two choices of how to develop the rules. He can have the shell fill-in the if part and only enter the solution, or else enter both the "if" and the "then" part.

After all the rules have been entered the expert system must be tested to verify that it has been programmed correctly. TIMM has two utilities for testing the expert system. The simplest is just to enter problems and evaluate the solutions. Two other utilities are to have the shell check the "consistency", and "completeness" of the rules. Both of these are very useful utilities, since for large expert systems it would be difficult or even impossible to perform these functions by "hand". Another convenient utility is the "generalize" command, in which the shell combines similar rules into a single rule and then asks the knowledge engineer to verify that the new rule is correct.

Since TIMM allows separate expert system modules to be chained together, large systems can be separated into small individual modules. For TIMM to access external programs at least one driver program, and probably several, must be written in FORTRAN; these programs get the data to be passed to TIMM, call TIMM, and then pass the data to TIMM. A driver program is also required if TIMM is to interact with databases.[10,14]

D. EXSYS

EXSYS is a rule-based shell which uses if-then-else type rules. The basic reasoning process for EXSYS is backward chaining. The shell starts with a rule containing choice #1. Based on the premise from this rule, a search is made for a fact that verifies this premise. If such a fact is not found, a search is made for a rule that can be used to infer the fact, or else the user is asked to supply the answer. This cycle is repeated until all the choices have been evaluated. In addition to backward chaining, EXSYS offers several optional reasoning processes. These are finalpass, forward, nobackward, nobackward and forward. Finalpass instructs the shell to use backward

chaining but after evaluating all conditions then execute any rules not previously used. Forward causes the shell to execute the rules in numeric order but to use backward chaining to determine the values of any unknown facts. If a rule has previously been executed then it is not executed again, since the state of the condition is already known. For nobackward, the shell uses backward chaining except for determining unknown facts; the user is queried if a fact is not known. Finally, nobackward and forward combines the two options instructing the shell to execute the rules in numeric order and to query the user if a fact is not known.

Table 3. Decision Structure for a TIMM Expert System

DECISION:

THE TYPE OF PROPELLANT TO USE
Choices:
AMM PERC
MET AP
SB
DB
TB
INERT NIT

FACTORS:

APPLICATION
Type of values: Linearly-Ordered Descriptive Phrases
Values:
ROCKET
SMALL CAL GUN
LARGE CAL GUN

TIMEFRAME
Type of values: Linearly-Ordered Descriptive Phrases
Values:
SHORT
MEDIUM LONG
LONG

COST
Type of values: Linearly-Ordered Descriptive Phrases
Values:
CHEAP
MODERATE
EXPENSIVE

VERBOSE VERSIONS

None

HELP INFORMATION

None

Table 4. Examples of TIMM's Rules

RULE 21

```
   FACTOR           VALUE
 If:
    ENERGY              IS MEDIUM
    BURN RATE             IS VERY HIGH

 Then:
    THE TYPE OF PROPELLANT TO USE IS EXPERIMTL(10)
                    CAT DB(45)
                    CAT NIT(45)
```

RULE 38

```
 If:
    APPLICATION            IS >= SMALL CAL GUN
    ENERGY              IS VERY HIGH

 Then:
    THE TYPE OF PROPELLANT TO USE IS ENER NIT(30)
                    EXPERIMTL(70)
```

EXCLUSIONARY RULE

RULE 1S

```
 If:
    APPLICATION            IS SMALL CAL GUN
    PROCESS TYPE           IS CAST CURE

 Then:
    THE TYPE OF PROPELLANT TO USE IS #
```

In addition to the if-then-else parts of the rule, a note and reference can also be associated with the rule. This is convenient for documentation, and it also makes more information available to the user. He can ask to see the note and reference, if desired. This gives him the option of obtaining a more detailed explanation, and suggests a source for additional information. A typical EXSYS if-then-else rule is shown in Table 5. A special editor is supplied for use in developing the rules. As factors are used in the rules, the editor keeps track of the factors and their allowed values. This is convenient since the factors and their values can be recalled and selected; they do not have to be remembered.

This shell supports mathematical functions including most trigonometric functions. Also, external programs can easily be called from within the shell. The external program is executed by calling the function "run (program name)". There are two options for passing data to the external program: (1) The data can be passed as a command-line argument. (2) The data is written to a disk file which is then read by the external program. The data is passed back to the shell by the second method; the external program writes the data

to disk and then the shell reads the data. A special provision is set up for reading reports generated from databases. The data from the database is written to a disk file with "end" used to separate the database records. The shell searches through the disk file reading the data as needed. An example of this type of disk file is shown in Table 6.

Table 5. Examples of EXSYS Rules

RULE NUMBER: 1

IF:
The oxidizer cost is unlimited

THEN:
[DESIRED OXIDIZER COST] IS GIVEN THE VALUE UNLIMITED

NOTE:

The knowledge-base is requesting information on the maximum allowable oxidizer cost. This cost will be used in screening the database for candidate oxidizers. Usually, the oxidizer cost controls the materials cost of the formulation. Enter "unlimited" if you do not want cost to be a criterion.

REFERENCE:

NONE

Table 6. Sample Diskfile for Accessing by EXSYS

```
V1  HTPB
V2 15.14
V3 0.899
V4 T
END
V1  CAB
V2 222.000
V3 120.000
V4  F
END
```

EXSYS also offers two utilities to increase the execution speed of the expert system. The utility "FASTER" increases execution time by organizing the rules to achieve an optimum order. The other utility available is "SHRINK" which organizes the text for rapid access and removes all unnecessary text, unused formulas and variables.[11]

E. INSIGHT 2+

INSIGHT 2+ (ver 1.3b) is another rule-based shell. It uses a computer programming language called Production Rule Language (PRL) which was specifically developed for creating expert systems. The reasoning process for this shell is the same as EXSYS, i.e., backward chaining. The shell starts

with a goal, then searches for a rule containing a premise that supports the goal. Once the rule is found, a search is made for a fact that verifies the premise. If such a fact is not found, a search is made for a rule that can be used to infer the fact or else the user is asked to supply the answer. This cycle is repeated until the goal is proved or disproved.

PRL encompasses about 65 keywords which include most mathematical functions (for example log, cos, sin, tan, abs, etc.). The expert system is developed just as one would develop a computer program. Once the expert system is complete, it is compiled, and then executed. Table 7 shows a small part of a source code listing of an expert system written in PRL. From the listing one can see the general structure of an expert system. Variables must be declared at the beginning of the program. The language supports four types of variables (numeric, string, boolean, and object-attribute). The goal(s) of the expert system must be defined at the beginning of the program. The main body of the program uses if-then-else rules. The expert system can be programmed to continue until all possibilities are exhausted; this is useful for interacting with databases. Since the expert system is written in PRL, one has considerable flexibility and freedom in developing the expert system.

Along with the PRL compiler, a database-accessing PASCAL-type compiler (titled DBPAS) is supplied with the shell package. This language can be used for interactions between expert systems and dBASEIII database files. Programs can be written in this language for managing dBASEIII databases. DBPAS was written so that data could be easily passed back and forth between the expert systems and the DBPAS program. Shown in Table 8 is a source code listing of a simple external DBPAS program (fetchbin) that "fetches" a binder from a database of binders and returns the name of the binder and three of its thirteen tabulated properties (cost, class, and inert/energetic designation) to the main program. For both DBPAS programs and programs written in other languages, there are two choices for passing data: the data can be passed via memory, or via a disk file. For accessing DBPAS programs "call program name" is used to invoke the program, while for programs written in other languages "activate program name" is used. A sample PRL program giving an example of interfacing to both a DBPAS and an TURBO PASCAL program is shown in Table 9. Supplied with the expert system shell package are some example programs of passing data between the expert system and TURBO PASCAL programs. In addition, there are supplied TURBO PASCAL procedures for passing data back and forth to the shell which can be used without any modifications. Also supplied is an example expert system which showed how INSIGHT 2+ can be programmed to use forward chaining as the reasoning process. An editor was supplied for use in developing the programs, but most word processors can be used.[12]

III. DISCUSSION

Of the five expert systems evaluated, three were example-based, while two were rule-based. Except for TIMM, all the other shell's inference engines supported forward chaining and backward chaining. All the shells supported chaining of modules so that expert systems could be written in small sections and then chained together. The three example-based expert system shells did not support mathematical functions.

KDS is an example-based shell. The version evaluated did not support mathematical functions. The major problems with KDS were the advanced

Table 7. Partial INSIGHT 2+ Source Code Listing

```
TITLE preliminary screening DISPLAY
press any key to begin

VARIABLES:

SHARED STRING oxidizer name
AND STRING desired oxidizer type
AND STRING desired binder type
AND STRING desired plasticizer type
AND STRING desired oxidizer cost

SHARED SIMPLEFACT desire a specific oxidizer
AND SIMPLEFACT desire a specific oxidizer type
AND SIMPLEFACT desire a specific binder
AND SIMPLEFACT desire a specific binder type

SHARED NUMERIC oxidizer number
AND NUMERIC binder number
AND NUMERIC plasticizer number

INIT oxidizer number = 1
INIT binder number = 1
INIT plast number = 1

FORGET desire a specific binder type
FORGET desire a specific plast
FORGET desire a specific plast type

GOAL:

1. have evaluated propellant

MAIN BODY:

RULE check for completeness
IF have evaluated desired oxidizer
!AND have evaluated desired binder
!AND have evaluated desired plast
THEN have evaluated propellant
AND DISPLAY answer
AND CHAIN OXIDC
!
RULE oxidizer name
IF desire a specific oxidizer
AND desired oxidizer name IS rdx
THEN have evaluated desired oxidizer
AND oxidizer name := 'RDX'
```

Table 8. DBPAS Source Code Listing

```
PROGRAM OXFETCH (RECEIVE INDEX : integer;
         RETURN COST : REAL;
             NAME, CLASS : STRING(25);
             ENERGETIC : BOOLEAN;
             STATUS : INTEGER);

VAR
 LAST : INTEGER;
 A   : CHAR;
OXIDIZER : RECORD
  NAME : STRING(25);
  CLASS : STRING(25);
  MOL_FOR : STRING(15);
  MOL_WGHT : REAL;
  HT_OF_FORM : REAL;
  ENERGETIC : BOOLEAN;
  DENSITY :REAL;
  IMP_SENS :REAL;
  BURN_RATE : REAL;
  COST : REAL;
  PARTICLE_S :REAL;
 END;

BEGIN
 OPEN (OXIDIZER, 'OXIDIZER');
 STATUS := 1;
 LAST := SIZE(OXIDIZER);
IF INDEX <= LAST THEN BEGIN
  GOTO (INDEX, OXIDIZER);
  STATUS :=0;
  COST := OXIDIZER.COST;
  NAME := OXIDIZER.NAME;
  CLASS := OXIDIZER.CLASS;
  ENERGETIC :=OXIDIZER.ENERGETIC;
 END;
 CLOSE (OXIDIZER);
END;
```

Table 9. Example of Accessing External Programs

```
TITLE oxidizer screening

STRING oxidizer name
AND oxidizer type
AND desired type

SIMPLEFACT energetic oxidizer
AND desire energetic

NUMERIC cost of oxidizer

INIT oxidizer number = 1

FORGET have evaluated oxidizer cost
FORGET have evaluated oxidizer
FORGET have saved oxidizer
FORGET have a component
FORGET get next oxidizer

SUPPRESS ALL

EXHAUSTIVE ALL

1. have evaluated oxidizer

RULE initialize datafile
ACTIVATE initoxid.com
THEN initialized oxidizer datafile

RULE Get entry from the oxidizer database
CALL OXFETCH
SEND oxidizer number !record number
RETURN cost of oxidizer
RETURN oxidizer name
RETURN oxidizer type
RETURN energetic oxidizer
RETURN oxidizer eof
IF oxidizer eof = 0
AND have saved oxidizer
THEN have evaluated oxidizer
AND oxidizer number := oxidizer number + 1
AND CYCLE
ELSE STOP

!
!unlimited >100.00
RULE Can oxidizer cost be unlimited
IF oxidizer cost IS unlimited
THEN have evaluated oxidizer cost
```

programming required for interfacing to external programs, and no provisions for searching databases. We found the system awkward and difficult to use. Correcting information was difficult; we usually ended up deleting all and starting over from scratch.

1st-CLASS is another example-based shell. It uses a spreadsheet-type format for the user interface. 1st-CLASS uses for its rule a single challenge tree which is developed based on the examples entered. External programs can be accessed directly from within the shell.

TIMM is the last example-based expert system shell evaluated. The user is expected to enter the factors and the associated values. The rules are developed from this information. There was also supplied with the system some utilities which could be used for debugging and testing. If the expert system is to interact with external programs or access databases, then a FORTRAN program must be written to interface the expert system to the external program. For non-programmers, this might pose a problem.

EXSYS is a rule-based expert system shell. The rules are the if-then-else type format. The rules are very structured and are difficult to work with. A function is supplied within the shell for interfacing to external programs. The shell supports most mathematical and trigonometric functions.

The last expert system shell evaluated is INSIGHT 2+ which is a rule-based expert system shell. This shell uses a compiler-based programming language called PRL. This language supports most mathematical and trigonometric functions. With PRL is supplied a pascal-type language for searching dBASEIII databases. There are also procedures supplied for passing data to and from TURBO PASCAL programs.

IV. CONCLUSION

Of the five shells evaluated, INSIGHT 2+ offers the best features for the development of the propellant formulation expert system. The three distinct advantages that INSIGHT 2+ offers are the relative ease of interfacing to TURBO PASCAL programs, accessing dBASEIII databases, and that the shell uses a programming language (though a very rudimentary language), which gives greater flexibility.

Expert system shells, depending on whether they are example-based or rule-based, each offer distinct advantages and disadvantages. The example-based systems offer the advantage of being easier to learn to use. All that is necessary is to enter specific examples. The shell develops the rules based on these examples. Another advantage that these shells offer is that they can reveal correlations in the data, since they develop the rules. Thus, in some cases, it may be valuable to analyze data with an example-based system in order to deduce rules that can then be incorporated into a rule-based system of greater flexibility and power. In addition, it has recently been shown that certain example-based shells like TIMM can even be used to perform "pattern recognition" analysis of experimental data.[15] Rule-based systems are more difficult to learn to use, since the rules have to be entered as opposed to just entering examples. However this offers the knowledge engineer increased control and power over how the expert system is written.

Whether the knowledge engineer decides to use an example-based or rule-based system depends on the type of application and his knowledge and experience in computer programming. One point to note is that all the shells could be interfaced to external programs. However each required varying degrees of programming skills to write the interfacing procedures. The commercial development of expert system shells is an evolving process; new shells are constantly being marketed and the shells already on the market are being updated, with new versions being offered continuously.

REFERENCES

1. J. McDermott, "A Rule-Based Configurer of Computer Systems," Technical Report CMU-CS-80-119, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, 1980.

2. R.K. Lindsay, B.G. Buchanan, E.A. Feigenbaum, and J. Lederberg, Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project, New York: McGraw-Hill, 1980.

3. E.H. Shortliffe, Computer-Based Medical Consultation: MYCIN, New York: American Elsevier, 1976.

4. F. Hayes-Roth, D.A. Waterman, and D.B. Lenat, Building Expert Systems, Addison-Wesley Publishing Corp., Reading MA, 1983.

5. P. Harmon and D. King, Expert Systems, Artificial Intelligence in Business, John Wiley & Sons, Inc., 1985.

6. "Expert Systems", Software News, pp. 38-41, July 1987.

7. D.A. Waterman, A Guide to Expert Systems, Addison-Wesley Publishing Company, 1986.

8. KDS Development System, KDS Corp., Wilmette, IL, 1985.

9. 1st-CLASS Instruction Manual Rel. 2.0, Programs in Motion Inc., Wayland, MA, 1985.

10. TIMM User's Manual, General Research Corp., Mclean, VA, 1985.

11. Expert System Development Package, EXSYS, Inc., Albuquerque, NM, 1985.

12. INSIGHT 2+ Reference Manual, Level Five Research, Inc., Indialantic, FL, 1986.

13. J.R. Quinlan, R.S. Michalski, J.G. Carbonell, T.M. Mitchell, Eds.; Machine Learning, An Artificial Intelligence Approach, Tioga Publishing Co.; Palo Alto, CA, 1983.

14. P. McWhite and P. Gerald, TIMM: A FORTRAN-Based Expert Systems Applications Generator, General Reasearch Corp., 7655 Old Springhouse Road, Mclean, VA, February 1985.

15. M.P. Derde, L. Buydens, C. Guns, and D.L. Massart, "Comparison of Rule-Building Expert Systems with Pattern Recognition for the Classification of Analytical Data," Anal. Chem., Vol. 59, pp. 1868-1871, 1987.

16. P.E. Lehner and S.W. Barth, "Expert Systems on Microcomputers," Expert Systems, Vol. 2, No. 4, October 1985.

17. M. Richer, "Five Commercial Expert Systems Tools: An Evaluation," The Artificial Intelligence Report, Vol. 2, No. 8, August 1985.

18. "Directory of Microcomputer-Based Software for Expert Systems Work," Expert Systems, Vol. 2, No. 4, October 1985.

19. M. Williamson, "Made-to-Order Mentors," PC Products, pp. 55-60, December 1985.

20. P. Kinnucan, "Software Tools Speed Expert System Development," High Technology, pp. 16-20, March 1985.

APPENDIX A.

LISTING OF PERSONAL COMPUTER-BASED EXPERT SYSTEM SHELLS

APPENDIX A. LISTING OF PERSONAL COMPUTER-BASED EXPERT SYSTEM SHELLS[5,15-19]

1st-CLASS
Programs in Motion Inc.
10 Sycamore Road
Wayland, MA 01778
617-879-9650
750.00

AION DEVELOPMENT SYSTEM-PC
Aion Corp
101 University Avenue
Palo Alto, CA 94303
415-328-9595
7000.00

ARITY ES DEVELOPMENT PACK
Arity Corp
358 Baker Avenue
Concord, MA 01742
617-371-1243
300.00

APES: EXPERT SYSTEM SHELL
Programming Logic Systems
312 Crescent Drive
Milford, CT 06460
203-877-7988
250.00

ARBORIST
Texas Instruments
12501 Research Blvd. MS 2244
Austin, TX 78769
800-527-3500
595.00

ART
Inference Corp.
5300 W. Century Blvd.
Los Angeles, CA 90045
213-417-7997
?

DECISION AIDE: DECISION SUPPORT
Kepner-tregoe, Inc.
P.O. BOX 704
Princeton, NJ 08542
609-921-2806
250.00

DECISON MAKER: DECISION SUPPORT
Alamo Learning Systems
1850 Mt. Diablo Blvd., Suite 500
Walnut Creek, CA 94596
415-930-8521
250.00

DUCK
Smart Systems Tech.
6870 Elm Street
McLean, VA 22101
703-448-8562
?

ERS: EXPERT SYSTEM SHELL
PAR Technology Corp.
220 Seneca Turnpike
New Hartford, NY 13413
?

ESIE
Lightwave Consultants
P.O. Box. 290539
Tampa, FL 33617
145.00

ESP ADVISOR
Expert Systems International
1150 First Avenue
King of Prussia, PA 19406
215-337-2300
900.00

EXPERT CHOICE: DECISION SUPPORT
Decision Support Software
1300 Vincent Place
Mclean, VA 22101
703-442-7900
500.00

EXPERT EDGE
Human Edge Software
2445 Faber Place
Palo Alto, CA 94303
415-493-1593
800.00

EXPERT-2
Miller Microcomputer Services
61 Lakeshore Road
Natick, MA 01760
617-653-6136
1250.00

EXSYS
EXSYS, Inc.
P.O. BOX 75158
Albuqerque, NM 87194
505-836-6676
400.00

ENVISAGE
System Designers Software, Ltd.
444 Washington St., Suite 407
Woburn, MA 01801
617-935-8009
?

GURU
MBDS (Micro Data Base Systems)
P.O. BOX 248
Lafayette, IN 47902
317-463-2581
3000.00

INFERENCE MANAGER: EXPERT SYSTEM SHELL
Intellignet Terminals Ltd.
15 Canal Street
Oxford, OX26BH United Kingdom
1000.00

INSIGHT 2+
Level Five Research Inc.
4980 S. Highway A1-A
Melbourne Beach, FL 32591
305-792-9046
500.00

K:BASE
Gold Hill Computers
163 Harvard Street
Cambridge, MA 02139
617-492-2071
?

KDS DEVELOPMENT SYSTEMS
KDS Corp.
934 Hunter Road
Wilmette, IL 90091
312-251-2621
800.00

KEE: EXPERT SYSTEM SHELL
Intellicorp
1975 El Camino Real W.
Mountain View, CA 94040
415-965-5500
?

KES II
SA & E Inc.
1500 Wilson Blvd.
Arlington, VA 22209
703-276-7910
4000.00

KNOWOL: EXPERT SYSTEM SHELL
Intelligent Machines Co.
3813 N. 14TH St.
Arlington, VA 22201
703-528-9136
?

LIGHTYEAR: DECISION SUPPORT
Lightyear, Inc.
1333 Lawrence Expwy., Bldg. 210
Santa Clara, CA 95051
408-985-8811
500.00

M1 VER 1.3
Teknowledge
525 University Avenue
Palo Alto, CA 943011
415-327-6600
5000.00

MICRO IN-ATE
Automated Reasoning Corp.
290 W. 12TH St., Suite 1D
New York, NY 10014
212-206-6331
5000.00

MICRO-PS
Software Architecture & Engineering
1500 Wilson Blvd.
Arlington, VA 22209
703-276-7910
?

MICROEXPERT: EXPERT SYSTEM SHELL
McGraw-Hill
1221 Avenue of the Americas
New York, NY 08520
800-628-0004
50.00

NEXPERT OBJECT FOR AT
Neuron Data
444 High Street
Palo Alto, CA 94301
415-321-4488
?

OPS5+
Artelligence, Inc.
14902 Preston Rd. Suite 212-252
Dallas, TX 75240
214-437-0631
?

OPS83: EXPERT SYSTEM SHELL
Production Systems Technology
642 Gettysburg Street
Pittsburgh, PA 15206
412-362-3117
?

PERSONAL CONSULTANT PLUS
Texas Instrument Inc.
P.O. BOX 809063 H-809
Dallas, TX 75380
214-680-5001
3000.00

PRODIGY
Artelligence Inc.
14902 Preston Rd. Suite 212
Dallas, TX 75240
214-437-0361
450.00

QTIM[illegible]
Mom [illegible]
Two Northsi[illegible] 75
Atlanta, GA 30318
404-351-2902
700.00

RULEMASTER
Radian Corp.
P.O. BOX 9948
Austin, TX 78766
512-454-4797
1000.00

REVEAL
Mcdonnell Douglas
20705 Valley Green Drive
Cupertino, CA 95014
408-446-6324
2000.00

SAGE
Systems Designers Software, Inc.
444 Washington Street, Suite 407
Woburn, MA 01801
617-935-8009
?

SERIES PC: EXPERT SYSTEM SHELL
SRI International ACS Division
333 Ravenswood Avenue
Menlo Park, CA 94025
415-859-2859
15000.00

SMALL-X
RK Software
P.O. Box 2085
West Chester, PA 19380
215-436-4570
225.00

THE IDEA GENERATOR: EXPERIENCE IN SOFTWARE
2039 Sattuck Avenue, Suite 401
Berkeley, CA 94704
415-644-0694
200.00

TIMM
General Research Corp.
7655 Old Springhouse Rd.
Mclean, VA 22102
703-893-5900
9500.00

TOPSCI: EXPERT SYSTEM SHELL
Dynamic Master Systems Inc.
P.O. BOX 566456
Atlanta, GA 30356
404-565-0771
175.00

WISDOM PX
Software Intelligence Lab Inc.
50 Broad St. 10TH FL
New York, NY 10040
516-589-1676
2050.00

XI
Portable Software, Inc.
650 Bair Island Road, Suite 204
Redwood City, CA   94063
415-367-6244
?

XSYS
California Intelligence
912 Powell Street #8
San Francisco, CA   94108
415-391-4846
1000.00

DISTRIBUTION LIST

No. Of Copies	Organization
12	Administrator Defense Technical Info Center ATTN: DTIC-FDAC Cameron Station, Bldg. 5 Alexandria, VA 22304-6145
1	HQ DA DAMA-ART-M Washington, DC 20310
1	Commander US Army Materiel Command ATTN: AMCDRA-ST 5001 Eisenhower Avenue Alexandria, VA 22333-0001
10	C.I.A. OIR/DB/Standard GE47 HQ Washington, DC 20505
1	Commander US Army ARDEC ATTN: SMCAR-MSI Dover, NJ 07801-5001
1	Commander US Army ARDEC ATTN: SMCAR-TDC Dover, NJ 07801
1	Commander US AMCCOM ARDEC CCAC Benet Weapons Laboratory ATTN: SMCAR-CCB-TL Watervliet, NY 12189-4050
1	US Army Armament, Munitions and Chemical Command ATTN: AMSMC-IMP-L Rock Island, IL 61299-7300
1	Commander US Army Aviation Systems Command ATTN: AMSAV-ES 4300 Goodfellow Blvd. St. Louis, MO 63120-1798

No. Of Copies	Organization
1	Director US Army Aviation Research and Technology Activity Ames Research Center Moffett Field, CA 94035-1099
4	Commander US Army Research Office ATTN: R. Ghirardelli D. Mann R. Singleton R. Shaw P.O. Box 12211 Research Triangle Park, NC 27709-2211
1	Commander US Army Communications - Electronics Command ATTN: AMSEL-ED Fort Monmouth, NJ 07703
1	Commander CECOM R&D Technical Library ATTN: AMSEL-IM-L, Reports Section B.2700 Fort Monmouth, NJ 07703-5000
2	Commander Armament R&D Center US Army AMCCOM ATTN: SMCAR-LCA-G, D.S. Downs J.A. Lannon Dover, NJ 07801
1	Commander Armament R&D Center US Army AMCCOM ATTN: SMCAR-LC-G, L. Harris Dover, NJ 07801
1	Commander Armament R&D Center US Army AMCCOM ATTN: SMCAR-SCA-T, L. Stiefel Dover, NJ 07801

DISTRIBUTION LIST

No. Of Copies	Organization
1	Commander US Army Missile Command Research, Development and Engineering Center ATTN: AMSMI-RD Redstone Arsenal, AL 35898
1	Commander US Army Missile and Space Intelligence Center ATTN: AMSMI-YDL Redstone Arsenal, AL 35898-5000
2	Commander US Army Missile Command ATTN: AMSMI-RK, D.J. Ifshin W. Wharton Redstone Arsenal, AL 35898
1	Commander US Army Missile Command ATTN: AMSMI-RKA, A.R. Maykut Redstone Arsenal, AL 35898-5249
1	Commander US Army Tank Automotive Command ATTN: AMSTA-TSL Warren, MI 48397-5000
1	Director US Army TRADOC Systems Analysis Center ATTN: ATOR-TSL White Sands Missile Range, NM 88002-5502
1	Commandant US Army Infantry School ATTN: ATSH-CD-CS-OR Fort Benning, GA 31905-5400
1	Commander US Army Development and Employment Agency ATTN: MODE-ORO Fort Lewis, WA 98433-5000

No. Of Copies	Organization
1	Office of Naval Research Department of the Navy ATTN: R.S. Miller, Code 432 800 N. Quincy Street Arlington, VA 22217
1	Commander Naval Air Systems Command ATTN: J. Ramnarace, AIR-54111C Washington, DC 20360
2	Commander Naval Ordnance Station ATTN: C. Irish P.L. Stang, Code 515 Indian Head, MD 20640
1	Commander Naval Surface Weapons Center ATTN: J.L. East, Jr., G-23 Dahlgren, VA 22448-5000
2	Commander Naval Surface Weapons Center ATTN: R. Bernecker, R-13 G.B. Wilmot, R-16 Silver Spring, MD 20902-5000
1	Commander Naval Weapons Center ATTN: R.L. Derr, Code 389 China Lake, CA 93555
2	Commander Naval Weapons Center ATTN: Code 3891, T. Boggs K.J. Graham China Lake, CA 93555
5	Commander Naval Research Laboratory ATTN: M.C. Lin J. McDonald E. Oran J. Shnur R.J. Doyle, Code 6110 Washington, DC 20375

DISTRIBUTION LIST

No. Of Copies	Organization
1	Commanding Officer Naval Underwater Systems Center Weapons Dept. ATTN: R.S. Lazar/Code 36301 Newport, RI 02840
1	Superintendent Naval Postgraduate School Dept. of Aeronautics ATTN: D.W. Netzer Monterey, CA 93940
4	AFRPL/DY, Stop 24 ATTN: R. Corley R. Geisler J. Levine D. Weaver Edwards AFB, CA 93523-5000
1	AFRPL/MKPB, Stop 24 ATTN: B. Goshgarian Edwards AFB, CA 93523-5000
1	AFOSR ATTN: J.M. Tishkoff Bolling Air Force Base Washington, DC 20332
1	AFATL/DOIL (Tech Info Center) Eglin AFB, FL 32542-5438
1	Air Force Weapons Laboratory AFWL/SUL ATTN: V. King Kirtland AFB, NM 87117
1	NASA Langley Research Center Langley Station ATTN: G.B. Northam/MS 168 Hampton, VA 23365
4	National Bureau of Standards ATTN: J. Hastie M. Jacox T. Kashiwagi H. Semerjian US Department of Commerce Washington, DC 20234

No. Of Copies	Organization
1	OSD/SDIO/UST ATTN: L.H. Caveny Pentagon Washington, DC 20301-7100
1	Aerojet Solid Propulsion Co. ATTN: P. Micheli Sacramento, CA 95813
1	Applied Combustion Technology, Inc. ATTN: A.M. Varney P.O. Box 17885 Orlando, FL 32860
2	Applied Mechanics Reviews The American Society of Mechanical Engineers ATTN: R.E. White A.B. Wenzel 345 E. 47th Street New York, NY 10017
1	Atlantic Research Corp. ATTN: M.K. King 5390 Cherokee Avenue Alexandria, VA 22314
1	Atlantic Research Corp. ATTN: R.H.W. Waesche 7511 Wellington Road Gainesville, VA 22065
1	AVCO Everett Rsch. Lab. Div. ATTN: D. Stickler 2385 Revere Beach Parkway Everett, MA 02149
1	Battelle Memorial Institute Tactical Technology Center ATTN: J. Huggins 505 King Avenue Columbus, OH 43201
1	Cohen Professional Services ATTN: N.S. Cohen 141 Channing Street Redlands, CA 92373

DISTRIBUTION LIST

No. Of Copies	Organization
1	Exxon Research & Eng. Co. ATTN: A. Dean Route 22E Annandale, NJ 08801
1	Ford Aerospace and Communications Corp. DIVAD Division Div. Hq., Irvine ATTN: D. Williams Main Street & Ford Road Newport Beach, CA 92663
1	General Applied Science Laboratories, Inc. ATTN: J.I. Erdos 425 Merrick Avenue Westbury, NY 11590
1	General Electric Armament & Electrical Systems ATTN: M.J. Bulman Lakeside Avenue Burlington, VT 05401
1	General Electric Company 2352 Jade Lane Schenectady, NY 12309
1	General Electric Ordnance Systems ATTN: J. Mandzy 100 Plastics Avenue Pittsfield, MA 01203
2	General Motors Rsch Labs Physics Department ATTN: T. Sloan R. Teets Warren, MI 48090
2	Hercules, Inc. Allegany Ballistics Lab. ATTN: R.R. Miller E.A. Yount P.O. Box 210 Cumberland, MD 21501
1	Hercules, Inc. Bacchus Works ATTN: K.P. McCarty P.O. Box 98 Magna, UT 84044
1	Honeywell, Inc. Government and Aerospace Products ATTN: D.E. Broden/ MS MN50-2000 600 2nd Street NE Hopkins, MN 55343
1	IBM Corporation ATTN: A.C. Tam Research Division 5600 Cottle Road San Jose, CA 95193
1	IIT Research Institute ATTN: R.F. Remaly 10 West 35th Street Chicago, IL 60616
2	Director Lawrence Livermore National Laboratory ATTN: C. Westbrook M. Costantino P.O. Box 808 Livermore, CA 94550
1	Lockheed Missiles & Space Co. ATTN: George Lo 3251 Hanover Street Dept. 52-35/B204/2 Palo Alto, CA 94304
1	Los Alamos National Lab ATTN: B. Nichols T7, MS-B284 P.O. Box 1663 Los Alamos, NM 87545
1	National Science Foundation ATTN: A.B. Harvey Washington, DC 20550

DISTRIBUTION LIST

No. Of Copies	Organization
1	Olin Corporation Smokeless Powder Operations ATTN: V. McDonald P.O. Box 222 St. Marks, FL 32355
1	Paul Gough Associates, Inc. ATTN: P.S. Gough 1048 South Street Portsmouth, NH 03801
2	Princeton Combustion Research Laboratories, Inc. ATTN: M. Summerfield N.A. Messina 475 US Highway One Monmouth Junction, NJ 08852
1	Hughes Aircraft Company ATTN: T.E. Ward 8433 Fallbrook Avenue Canoga Park, CA 91303
1	Rockwell International Corp. Rocketdyne Division ATTN: J.E. Flanagan/HB02 6633 Canoga Avenue Canoga Park, CA 91304
4	Sandia National Laboratories Combustion Sciences Dept. ATTN: R. Cattolica S. Johnston P. Mattern D. Stephenson Livermore, CA 94550
1	Science Applications, Inc. ATTN: R.B. Edelman 23146 Cumorah Crest Woodland Hills, CA 91364
1	Science Applications, Inc. ATTN: H.S. Pergament 1100 State Road, Bldg. N Princeton, NJ 08540
3	SRI International ATTN: G. Smith D. Crosley D. Golden 333 Ravenswood Avenue Menlo Park, CA 94025
1	Stevens Institute of Tech. Davidson Laboratory ATTN: R. McAlevy, III Hoboken, NJ 07030
1	Textron, Inc. Bell Aerospace Co. Division ATTN: T.M. Ferger P.O. Box 1 Buffalo, NY 14240
1	Thiokol Corporation Elkton Division ATTN: W.N. Brundige P.O. Box 241 Elkton, MD 21921
1	Thiokol Corporation Huntsville Division ATTN: R. Glick Huntsville, AL 35807
3	Thiokol Corporation Wasatch Division ATTN: S.J. Bennett P.O. Box 524 Brigham City, UT 84302
1	TRW ATTN: M.S. Chou MSR1-1016 1 Parke Redondo Beach, CA 90278
1	United Technologies ATTN: A.C. Eckbreth East Hartford, CT 06108

DISTRIBUTION LIST

No. Of Copies	Organization
3	United Technologies Corp. Chemical Systems Division ATTN: R.S. Brown T.D. Myers (2 copies) P.O. Box 50015 San Jose, CA 95150-0015
2	United Technologies Corp. ATTN: R.S. Brown R.O. McLaren P.O. Box 358 Sunnyvale, CA 94086
1	Universal Propulsion Company ATTN: H.J. McSpadden Black Canyon Stage 1 Box 1140 Phoenix, AZ 85029
1	Veritay Technology, Inc. ATTN: E.B. Fisher 4845 Millersport Highway P.O. Box 305 East Amherst, NY 14051-0305
1	Brigham Young University Dept. of Chemical Engineering ATTN: M.W. Beckstead Provo, UT 84601
1	California Institute of Tech. Jet Propulsion Laboratory ATTN: MS 125/159 4800 Oak Grove Drive Pasadena, CA 91103
1	California Institute of Technology ATTN: F.E.C. Culick/ MC 301-46 204 Karman Lab. Pasadena, CA 91125
1	University of California, Berkeley Mechanical Engineering Dept. ATTN: J. Daily Berkeley, CA 94720

No. Of Copies	Organization
1	University of California Los Alamos Scientific Lab. P.O. Box 1663, Mail Stop B216 Los Alamos, NM 87545
2	University of California, Santa Barbara Quantum Institute ATTN: K. Schofield M. Steinberg Santa Barbara, CA 93106
2	University of Southern California Dept. of Chemistry ATTN: S. Benson C. Wittig Los Angeles, CA 90007
1	Case Western Reserve Univ. Div. of Aerospace Sciences ATTN: J. Tien Cleveland, OH 44135
1	Cornell University Department of Chemistry ATTN: T.A. Cool Baker Laboratory Ithaca, NY 14853
1	Univ. of Dayton Rsch Inst. ATTN: D. Campbell AFRPL/PAP Stop 24 Edwards AFB, CA 93523
1	University of Florida Dept. of Chemistry ATTN: J. Winefordner Gainesville, FL 32611
3	Georgia Institute of Technology School of Aerospace Engineering ATTN: E. Price W.C. Strahle B.T. Zinn Atlanta, GA 30332

DISTRIBUTION LIST

No. Of Copies	Organization
1	University of Illinois Dept. of Mech. Eng. ATTN: H. Krier 144MEB, 1206 W. Green St. Urbana, IL 61801
1	Johns Hopkins University/APL Chemical Propulsion Information Agency ATTN: T.W. Christian Johns Hopkins Road Laurel, MD 20707
1	University of Michigan Gas Dynamics Lab Aerospace Engineering Bldg. ATTN: G.M. Faeth Ann Arbor, MI 48109-2140
1	University of Minnesota Dept. of Mechanical Engineering ATTN: E. Fletcher Minneapolis, MN 55455
3	Pennsylvania State University Applied Research Laboratory ATTN: K.K. Kuo H. Palmer M. Micci University Park, PA 16802
1	Polytechnic Institute of NY Graduate Center ATTN: S. Lederman Route 110 Farmingdale, NY 11735
2	Princeton University Forrestal Campus Library ATTN: K. Brezinsky I. Glassman P.O. Box 710 Princeton, NJ 08540
1	Princeton University MAE Dept. ATTN: F.A. Williams Princeton, NJ 08544

No. Of Copies	Organization
1	Purdue University School of Aeronautics and Astronautics ATTN: J.R. Osborn Grissom Hall West Lafayette, IN 47906
1	Purdue University Department of Chemistry ATTN: E. Grant West Lafayette, IN 47906
2	Purdue University School of Mechanical Engineering ATTN: N.M. Laurendeau S.N.B. Murthy TSPC Chaffee Hall West Lafayette, IN 47906
1	Rensselaer Polytechnic Inst. Dept. of Chemical Engineering ATTN: A. Fontijn Troy, NY 12181
1	Stanford University Dept. of Mechanical Engineering ATTN: R. Hanson Stanford, CA 94305
1	University of Texas Dept. of Chemistry ATTN: W. Gardiner Austin, TX 78712
1	University of Utah Dept. of Chemical Engineering ATTN: G. Flandro Salt Lake City, UT 84112
1	Virginia Polytechnic Institute and State University ATTN: J.A. Schetz Blacksburg, VA 24061

DISTRIBUTION LIST

No. Of Copies	Organization
1	Commandant USAFAS ATTN: ATSF-TSM-CN Fort Sill, OK 73503-5600
1	F.J. Seiler Research Lab (AFSC) ATTN: S.A. Shakelford USAF Academy, CO 80840-6528
1	Commander Naval Weapons Center ATTN: Code 3851, A. Harper China Lake, CA 93555
1	University of New Hampshire Parsons Hall ATTN: S.A. Tomellini Durham, NH 03824-3598
1	Information Builders, Inc. ATTN: P.A. Ruhland Suite 404 111 West Port Plaza St. Louis, MO 63146
1	The Johns Hopkins Univeristy/APL Chemical Propulsion Info. Agency ATTN: T. Wilson Johns Hopkins Road Laurel, MD 20707

Aberdeen Proving Ground

Dir, USAMSAA
ATTN: AMXSY-D
AMXSY-MP, H. Cohen
Cdr, USATECOM
ATTN: AMSTE-SI-F
Cdr, CRDC, AMCCOM
ATTN: SMCCR-RSP-A
SMCCR-MU
SMCCR-SPS-IL

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. BRL Report Number________________________Date of Report__________

2. Date Report Received____

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.)________________

4. How specifically, is the report being used? (Information source, design data, procedure, source of ideas, etc.)______________________________

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided or efficiencies achieved, etc? If so, please elaborate.__________________________________

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.)

CURRENT ADDRESS

Name

Organization

Address

City, State, Zip

7. If indicating a Change of Address or Address Correction, please provide the New or Correct Address in Block 6 above and the Old or Incorrect address below.

OLD ADDRESS

Name

Organization

Address

City, State, Zip

(Remove this sheet, fold as indicated, staple or tape closed, and mail.)

FOLD HERE

Director
U.S. Army Ballistic Research Laboratory
ATTN: SLCBR-DD-T
Aberdeen Proving Ground, MD 21005-5066

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 12062 WASHINGTON, DC

POSTAGE WILL BE PAID BY DEPARTMENT OF THE ARMY

Director
U.S. Army Ballistic Research Laboratory
ATTN: SLCBR-DD-T
Aberdeen Proving Ground, MD 21005-9989

FOLD HERE